MARIA COLLEGE OF ENGINEERING AND TECHNOLOGY, ATTOOR

DEPARTMENT OF INFORMATION TECHNOLOGY

OPERATING SYSTEMS

2 MARKS QUESTIONS & ANSWERS

UNIT I

1.Define the functions of OS?

Main functions of OS:

- process management
- Memory management.
- Device or I/O management.
- File management.

Other functions of OS:

- Implementing the user interface
- Sharing network among users
- Preventing users from interfering with one another.
- Allocating the users to share data among users.
- scheduling resources among users.
- Facilitating input and output.
- Recovery from errors.
- Accounting for resource usage.
- Organizing data for secure and fast access.
- Facilitating parallel operations.
- Handling network connections.

2.What are the goals of OS?

Primary goal :

- To make a system convenient to use.
- It provides an environment , within this environment other programs can do useful work.

Secondary goal :

- Efficient operation of the computer system.

3.Compare Hard Real Time Operating System & Soft Real Time Operating System ?

Hard Real Time Operating System	Soft Real Time Operating System
1.Time delays are not allowed	1.Time delays are allowed
2.It assures critical tasks must completed on time	2.Very critical tasks gets more priority ever other tasks and retains it until it completes.
3.All delays in the system should be bounded	3.All delays in the system should be bounded

4.What are Batch systems?

Batch systems are quite appropriate for executing largejobs that need little interaction. The user can submit jobs andreturn later for the results. It is not necessary to wait whilethe job is processed. Operators batched together jobs withsimilar needs and ran them through the computer as a group.

5.What is the advantage of Multiprogramming?

Multiprogramming increases CPU utilization by organizing jobs so that the CPU always has one to execute. Several jobs are placed in the main memory and the processor is switched from job to job as needed to keep several jobs advancing while keeping the peripheral devices in use. Multiprogramming is the first instance where the Operating system must make decisions for the users. Therefore they are fairly sophisticated.

6. What is an Interactive computer system?

Interactive computer system provides direct communication the operating system or to a program directly, using a keyboard or mouse ,and waits for immediate results.

7.What do you mean by Time-sharing systems?

Time-sharing or multitasking is a logical extension of multiprogramming. It allows many users to share the computer simultaneously. The CPU executes multiple jobs by switching among them, but the switches occur so frequently that the users can interact with each program while it is running.

8.What are multiprocessor systems & give their advantages?

Multiprocessor systems also known as parallel systems or tightly coupled systems are systems that have more than one processor in close communication, sharing the computer bus, the clock and sometimes memory & peripheral devices. Their main advantages are

- Increased throughput
- Economy of scale

- Increased reliability

9.What are the different types of multiprocessing?

Symmetric multiprocessing (SMP): In SMP each processor runs an identical copy of the Os & these copies communicate with one another as needed. All processors are peers. Examples are Windows NT, Solaris, Digital UNIX, OS/2 & Linux. Asymmetric multiprocessing: Each processor is assigned a specific task. A master processor controls the system; the other processors look to the master for instructions or predefined tasks. It defines a master-slave relationship. Example SunOS Version 4.

10.What is a process?

A process is a program in execution. It is the unit of work in a modern operating system. A process is an active entity with a program counter specifying the next instructions to execute and a set of associated resources. It also includes the process stack, containing temporary data and a data section containing global variables.

11.What is a process state and mention the various states of a process?

As a process executes, it changes state. The state of a process is defined in part by the current activity of that process. Each process may be in one of the following states:

- New
- Running
- Waiting
- Ready
- Terminated

12.What is process control block?

Each process is represented in the operating system by a process control block also called a task control block. It contains many pieces of information associated with a specific process. It simply acts as a repository for any information that may vary from process to process. It contains the following information:

- Process state
- Program counter
- CPU registers
- CPU-scheduling information
- Memory-management information
- Accounting information
- I/O status information

13.What are the use of job queues, ready queues & device queues?

As a process enters a system, they are put into a job queue. This queue consists of all jobs in the system. The processes that are residing in main memory and are ready & waiting to execute are kept on a list called ready queue. The list of processes waiting for a particular I/O device is kept in the device queue.

14.What is meant by context switch?

Switching the CPU to another process requires saving the state of the old process and loading the saved state for the new process. This task is known as context switch. The context of a process is represented in the PCB of a process.

15.What is a thread?

A thread otherwise called a lightweight process (LWP) is a basic unit of CPU utilization, it comprises of a thread id, a program counter, a register set and a stack. It shares with other threads belonging to the same process its code section, data section, and operating system resources such as open files and signals.

16.What are the benefits of multithreaded programming?

The benefits of multithreaded programming can be broken down into four major categories:
- Responsiveness
- Resource sharing
- Economy
- Utilization of multiprocessor architectures

17.Compare user threads and kernel threads.

<u>User threads</u>

User threads are supported above the kernel and are implemented by a thread library at the user level. Thread creation & scheduling are done in the user space, without kernel intervention. Therefore they are fast to create and manage blocking system call will cause the entire process to block

<u>Kernel threads</u>

Kernel threads are supported directly by the operating system .Thread creation, scheduling and management are done by the operating system. Therefore they are slower to create & manage compared to user threads. If the thread performs a blocking system call, the kernel can schedule another thread in the application for execution

18.What is the use of fork and exec system calls?

Fork is a system call by which a new process is created. Exec is also a system call, which is used after a fork by one of the two processes to replace the process memory space with a new program.

19.Define thread cancellation & target thread.

The thread cancellation is the task of terminating a thread before it has completed. A thread that is to be cancelled is often referred to as the target thread. For example, if multiple threads are concurrently searching through a database and one thread returns the result, the remaining threads might be cancelled.

20.What are the different ways in which a thread can be cancelled?

Cancellation of a target thread may occur in two different scenarios:

- Asynchronous cancellation:

One thread immediately terminates the target thread is called asynchronous cancellation.

- Deferred cancellation:

The target thread can periodically check if it should terminate, allowing the target thread an opportunity to terminate itself in an orderly fashion.

UNIT II

1.Define CPU scheduling.

CPU scheduling is the process of switching the CPU among various processes. CPU scheduling is the basis of multi programmed operating systems. By switching the CPU among processes, the operating system can make the computer more productive.

2.What is preemptive and non preemptive scheduling?

Under non preemptive scheduling once the CPU has been allocated to a process, the process keeps the CPU until it releases the CPU either by terminating or switching to the waiting state. Preemptive scheduling can preempt a process which is utilizing the CPU in between its execution and give the CPU to another process.

3.What is a Dispatcher?

The dispatcher is the module that gives control of the CPU to the process selected by the short-term scheduler. This function involves:

- Switching context
- Switching to user mode
- Jumping to the proper location in the user program to restart that program.

4.What is dispatch latency?

The time taken by the dispatcher to stop one process and start another running is known as dispatch latency.

5.What are the various scheduling criteria for CPU scheduling?

The various scheduling criteria are

- CPU utilization
- Throughput
- Turnaround time
- Waiting time
- Response time

6.Define throughput?

Throughput in CPU scheduling is the number of processes that are completed per unit time. For long processes, this rate may be one process per hour; for short transactions, throughput might be 10 processes per second.

7.What is turnaround time?

Turnaround time is the interval from the time of submission to the time of completion of a process. It is the sum of the periods spent waiting to get into memory, waiting in the ready queue, executing on the CPU, and doing I/O.

8.Define race condition.

When several process access and manipulate same data concurrently, then the outcome of the execution depends on particular order in which the access takes place is called race condition. To avoid race condition, only one process at a time can manipulate the shared variable.

9.What is critical section problem?

Consider a system consists of 'n' processes. Each process has segment of code called a critical section, in which the process may be changing common variables, updating a table, writing a file. When one process is executing in its critical section, no other process can allowed to execute in its critical section.

10.What are the requirements that a solution to the critical section problem must satisfy?

The three requirements are

- Mutual exclusion

- Progress
- Bounded waiting

11.Define entry section and exit section.

The critical section problem is to design a protocol that the processes can use to cooperate. Each process must request permission to enter its critical section. The section of the code implementing this request is the entry section. The critical section is followed by an exit section. The remaining code is the remainder section.

12.Give two hardware instructions and their definitions which can be used for implementing mutual exclusion.

- TestAndSet

```
boolean TestAndSet (boolean &target)
{
boolean rv = target;
target = true;
return rv;
}
```

- Swap

```
void Swap (boolean &a, boolean &b)
{
boolean temp = a;
a = b;
b = temp;
}
```

13.What is semaphores?

A semaphore 'S' is a synchronization tool which is an integer value that, apart from initialization, is accessed only through two standard atomic operations; wait and signal. Semaphores can be used to deal with the n-process critical section problem. It can be also used to solve various synchronization problems. The classic definition of 'wait'

```
wait (S)
{
while (S<=0)
;
S--;
}
```

The classic definition of 'signal'

```
signal (S)
{
S++;
```

}

14.Define busy waiting and spinlock.

When a process is in its critical section, any other process that tries to enter its critical section must loop continuously in the entry code. This is called as busy waiting and this type of semaphore is also called a spinlock, because the process while waiting for the lock.

15.Define deadlock.

A process requests resources; if the resources are not available at that time, the process enters a wait state. Waiting processes may never again change state, because the resources they have requested are held by other waiting processes. This situation is called a deadlock.

16.What is the sequence in which resources may be utilized?

Under normal mode of operation, a process may utilize a resource in the following sequence:

- Request: If the request cannot be granted immediately, then the requesting process must wait until it can acquire the resource.
- Use: The process can operate on the resource.
- Release: The process releases the resource.

17.What are conditions under which a deadlock situation may arise?

A deadlock situation can arise if the following four conditions hold simultaneously in a system:
a. Mutual exclusion
b. Hold and wait
c. No pre-emption

18.What is a resource-allocation graph?

Deadlocks can be described more precisely in terms of a directed graph called a system resource allocation graph. This graph consists of a set of vertices V and a set of edges E. The set of vertices V is partitioned into two different types ofnodes; P the set consisting of all active processes in the system and R the set consisting of all resource types in the system.

19.Define request edge and assignment edge.

A directed edge from process Pi to resource type Rj is denoted by Pi->Rj; it signifies that process Pi requested an instance of resource type Rj and is currently waiting for that resource. A directed edge from resource type Rj to process Pi is denoted by Rj->Pi, it signifies that an instance of resource type has been allocated to a process Pi. A directed edge Pi->Rj is called a request edge. A directed edge Rj->Pi is called an assignment edge.

20.What are the methods for handling deadlocks?

The deadlock problem can be dealt with in one of the three ways:

a. Use a protocol to prevent or avoid deadlocks, ensuring that the system will never enter a deadlock state.

b. Allow the system to enter the deadlock state, detect it and then recover.

c. Ignore the problem all together, and pretend that deadlocks never occur in the system.

21.What are a safe state and an unsafe state?

A state is safe if the system can allocate resources to each process in some order and still avoid a deadlock. A system is in safe state only if there exists a safe sequence. A sequence of processes <P1,P2,....Pn> is a safe sequence for the current allocation state if, for each Pi, the resource that Pi can still request can be satisfied by the current available resource plus the resource held by all the Pj, with j<i. if no such sequence exists, then the system state is said to be unsafe.

UNIT III

1.Define logical address and physical address.

An address generated by the CPU is referred as logical address. An address seen by the memory unit that is the one loaded into the memory address register of the memory is commonly referred to as physical address.

2.What is logical address space and physical address space?

The set of all logical addresses generated by a program is called a logical address space; the set of all physical addresses corresponding to these logical addresses is a physical address space.

3.What is the main function of the memory-management unit?

The runtime mapping from virtual to physical addresses is done by a hardware device called a memory management unit (MMU).

4.Define dynamic loading.

To obtain better memory-space utilization dynamic loading is used. With dynamic loading, a routine is not loaded until it is called. All routines are kept on disk in a relocatable load format. The main program is loaded into memory and executed. If the routine needs another routine, the calling routine checks whether the routine has been loaded. If not, the relocatable linking loader is called to load the desired program into memory.

5.Define dynamic linking.

Dynamic linking is similar to dynamic loading, rather that loading being postponed until execution time, linking is postponed. This feature is usually used with system libraries, such as language subroutine libraries. A stub is included in the image for each library-routine reference. The stub is a small piece of code that indicates how to locate the appropriate memory-resident library routine, or how to load the library if the routine is not already present.

6.What are overlays?

To enable a process to be larger than the amount of memory allocated to it, overlays are used. The idea of overlays is to keep in memory only those instructions and data that are needed at a given time. When other instructions are needed, they are loaded into space occupied previously by instructions that are no longer needed.

7.Define swapping.

A process needs to be in memory to be executed. However a process can be swapped temporarily out of memory to a backing store and then brought back into memory for continued execution. This process is called swapping.

8.What are the common strategies to select a free hole from a set of available holes?

The most common strategies are

a. First fit
b. Best fit
c. Worst fit

9.What do you mean by best fit?

Best fit allocates the smallest hole that is big enough. The entire list has to be searched, unless it is sorted by size. This strategy produces the smallest leftover hole.

10.What do you mean by first fit?

First fit allocates the first hole that is big enough. Searching can either start at the beginning of the set of holes or where the previous first-fit search ended. Searching can be stopped as soon as a free hole that is big enough is found.

11.What is virtual memory?

Virtual memory is a technique that allows the execution of processes that may not be completely in memory. It is the separation of user logical memory from physical memory. This separation provides an extremely large virtual memory, when only a smaller physical memory is available.

12.What is Demand paging?

Virtual memory is commonly implemented by demand paging. In demand paging, the pager brings only those necessary pages into memory instead of swapping in a whole process. Thus it avoids reading into memory pages that will not be used anyway, decreasing the swap time and the amount of physical memory needed.

13.Define lazy swapper.

Rather than swapping the entire process into main memory, a lazy swapper is used. A lazy swapper never swaps a page into memory unless that page will be needed.

14.What is a pure demand paging?

When starting execution of a process with no pages in memory, the operating system sets the instruction pointer to the first instruction of the process, which is on a non-memory resident page, the process immediately faults for the page. After this page is brought into memory, the process continues to execute, faulting as necessary until every page that it needs is in memory. At that point, it can execute with no more faults. This schema is called pure demand paging.

15.Define effective access time. (EAT)

Let p be the probability of a page fault (0£p£1). The value of p is expected to be close to 0; that is, there will be only a few page faults. The effective access time is Effective access time = (1-p) * ma + p * page fault time. ma : memory-access time

16.Define secondary memory.

This memory holds those pages that are not present in main memory. The secondary memory is usually a high speed disk. It is known as the swap device, and the section of the disk used for this purpose is known as swap space.

17.What is the basic approach of page replacement?

If no frame is free is available, find one that is not currently being used and free it. A frame can be freed by writing its contents to swap space, and changing the page table to indicate that the page is no longer in memory. Now the freed frame can be used to hold the page for which the process faulted.

18.What are the various page replacement algorithms used for page replacement?

- FIFO page replacement
- Optimal page replacement
- LRU page replacement
- LRU approximation page replacement
- Counting based page replacement

- Page buffering algorithm.

19.What are the major problems to implement demand paging?

The two major problems to implement demand paging is developing
a. Frame allocation algorithm
b. Page replacement algorithm

20.What is a reference string?

An algorithm is evaluated by running it on a particular string of memory references and computing the number of page faults. The string of memory reference is called a reference string.

UNIT IV

1.What is a file?

A file is a named collection of related information that is recorded on secondary storage. A file contains either programs or data. A file has certain "structure" based on its type. • File attributes: Name, identifier, type, size, location, protection, time, date • File operations: creation, reading, writing, repositioning, deleting, truncating, appending, renaming • File types: executable, object, library, source code etc.

2.List the various file attributes.

A file has certain other attributes, which vary from one operating system to another, but typically consist of these: Name, identifier, type, location, size, protection, time, date and user identification.

3.What are the various file operations?

The six basic file operations are

- Creating a file
- Writing a file
- Reading a file
- Repositioning within a file
- Deleting a file
- Truncating a file

4.What are the information associated with an open file?

Several pieces of information are associated with an open file which may be:

- File pointer
- File open count

- Disk location of the file
- Access rights

5. What are the different accessing methods of a file?

The different types of accessing a file are:

- Sequential access: Information in the file is accessed sequentially
- Direct access: Information in the file can be accessed without any particular order.
- Other access methods: Creating index for the file, indexed sequential access method (ISAM)

6. What is Directory?

The device directory or simply known as directory records information-such as name, location, size, and type for all files on that particular partition. The directory can be viewed as a symbol table that translates file names into their directory entries.

7. What are the operations that can be performed on a directory?

The operations that can be performed on a directory are

- Search for a file
- Create a file
- Delete a file
- Rename a file
- List directory
- Traverse the file system

8. What are the most common schemes for defining the logical structure of a directory?

The most common schemes for defining the logical structure of a directory

- Single-Level Directory
- Two-level Directory
- Tree-Structured Directories
- Acyclic-Graph Directories
- General Graph Directory

9. Define UFD and MFD.

In the two-level directory structure, each user has her own user file directory (UFD). Each UFD has a similar structure, but lists only the files of a single user. When a job starts the system's master file directory (MFD) is searched. The MFD is indexed by the user name or account number, and each entry points to the UFD for that user.

10. What is a path name?

A pathname is the path from the root through all subdirectories to a specified file. In a two-level directory structure a user name and a file name define a path name.

11. What are the various layers of a file system?

The file system is composed of many different levels. Each level in the design uses the feature of the lower levels to create new features for use by higher levels.

- Application programs
- Logical file system
- File-organization module
- Basic file system
- I/O control
- Devices

12.What are the structures used in file-system implementation?

Several on-disk and in-memory structures are used to implement a file system

a. On-disk structure include
· Boot control block
· Partition block
· Directory structure used to organize the files
· File control block (FCB)

b. In-memory structure include
· In-memory partition table
· In-memory directory structure
· System-wide open file table
· Per-process open table

13. What are the functions of virtual file system (VFS)?

It has two functions

a. It separates file-system-generic operations from their implementation defining a clean VFS interface. It allows transparent access to different types of file systems mounted locally.

b. VFS is based on a file representation structure, called a vnode. It contains a numerical value for a network-wide unique file .The kernel maintains one vnode structure for each active file or directory.

14.Define FCB (File control block).

FCB contains

1.File permissions
2. File Dates (create , access, write)
3. File owner , group, ACL
4.File size.
5. File data blocks

15.Define FAT (File Allocation Table).

An important variation on the linked allocation method is the use of FAT. A section of disk at the beginning of each partition is set aside to contain the table. The table has one entry for each disk block and is indexed by block number. The FAT is used much as is a linked list. The directory entry contains the block number of the first block of the file. The table entry indexed by that block number then contains the block number of the next block in the file. The chain continues until the last block, which has a special end-of file value as the table entry.

16.What is the operations for NFS protocol.

1. Searching for a file within a directory
2. Reading a set of directory entries
3. Manipulating links and directories
4. Accessing file attributes
5. Reading and writing files.

Path-Name Translation is done by breaking the path into component names and performing a separate NFS lookup call for every pair of component name and directory vnode. Once a mount point is crossed, every component lookup causes a separate RPC to the server. This expensive path-name-traversal scheme is needed, since each client has a unique layout of its logical name space, dictated by the mounts it performed.

18.What are immutable shared files?

1. Once a file is declared as shared by its creator, it cannot be modified.
2. An immutable file has two key properties. Its name may not be reused and its contents may not be altered.

19. What are the advantages of the variation of linked allocation that uses a FAT to chain together the blocks of a file?

A benefit is that random access time is improved, because the disk head can fine the location of any block by reading the information in the FAT.

20. What are the File Types?

1. Executable file
2. Object file

3. Source code
4. Batch files
5. Text Files
6. Library Files
7. Archive Files.

UNIT V

1. Define seek time and latency time.

The time taken by the head to move to the appropriate cylinder or track is called seek time. Once the head is at right track, it must wait until the desired block rotates under the read-write head. This delay is latency time.

2. What are the allocation methods of a disk space?

Three major methods of allocating disk space which are widely in use are

a. Contiguous allocation
b. Linked allocation
c. Indexed allocation

3. What are the advantages of Contiguous allocation?

The advantages are

a. Supports direct access
b. Supports sequential access
c. Number of disk seeks is minimal.

4. What are the drawbacks of contiguous allocation of disk space?

The disadvantages are

a. Suffers from external fragmentation
b. Suffers from internal fragmentation
c. Difficulty in finding space for a new file
d. File cannot be extended
e. Size of the file is to be declared in advance

5. What are the advantages of Linked allocation?

The advantages are
a. No external fragmentation
b. Size of the file does not need to be declared

6. What are the disadvantages of linked allocation?

The disadvantages are
a. Used only for sequential access of files.
b. Direct access is not supported
c. Memory space required for the pointers.
d. Reliability is compromised if the pointers are lost or damaged

7. What are the advantages of Indexed allocation?

The advantages are
a. No external-fragmentation problem
b. Solves the size-declaration problems.
c. Supports direct access

8. How can the index blocks be implemented in the indexed allocation scheme?

The index block can be implemented as follows
a. Linked scheme
b. Multilevel scheme
c. Combined scheme

9. Define rotational latency and disk bandwidth.

Rotational latency is the additional time waiting for the disk to rotate the desired sector to the disk head. The disk bandwidth is the total number of bytes transferred, divided by the time between the first request for service and the completion of the last transfer.

10. How free-space is managed using bit vector implementation?

The free-space list is implemented as a bit map or bit vector. Each block is represented by 1 bit. If the block is free, the bit is 1; if the block is allocated, the bit is 0.

11. Define buffering.

A buffer is a memory area that stores data while they are transferred between two devices or between a device and an application. Buffering is done for three reasons:

a. To cope with a speed mismatch between the producer and consumer of a data stream
b. To adapt between devices that have different data transfer sizes
c. To support copy semantics for application I/O

12. Define caching.

A cache is a region of fast memory that holds copies of data. Access to the cached copy is more efficient than access to the original. Caching and buffering are distinct functions, but sometimes a region of memory can be used for both purposes.

13. Define spooling.

A spool is a buffer that holds output for a device, such as printer, that cannot accept interleaved data streams. When an application finishes printing, the spooling system queues the corresponding spool file for output to the printer. The spooling system copies the queued spool files to the printer one at a time.

14. What are the various disk-scheduling algorithms?

The various disk-scheduling algorithms are
a. First Come First Served Scheduling
b. Shortest Seek Time First Scheduling
c. SCAN Scheduling
d. C-SCAN Scheduling
f. LOOK scheduling

15. What is low-level formatting?

Before a disk can store data, it must be divided into sectors that the disk controller can read and write. This process is called low-level formatting or physical formatting. Low-level formatting fills the disk with a special data structure for each sector. The data structure for a sector consists of a header, a data area, and a trailer.

16. What is the use of boot block?

For a computer to start running when powered up or rebootedit needs to have an initial program to run. This bootstrap program tends to be simple. It finds the operating system on the disk loads that kernel into memory and jumps to an initial address to begin the operating system execution. The full bootstrap program is stored in a partition called the boot blocks, at fixed location on the disk. A disk that has boot partition is called boot disk or system disk.

17. What is sector sparing?

Low-level formatting also sets aside spare sectors not visible to the operating system. The controller can be told to replace each bad sector logically with one of the spare sectors. This scheme is known as sector sparing or forwarding.

18. Differentiate RAID level 0 and RAID level 1 .

- RAID level 0 refers to disk arrays with striping at the level of blocks, but without and redundancy.
- RAID level 1 refers to disk mirroring.

19. What is meant by WORM disks.

- The data on read – write disks can be modified over and over.
- WORM - Write Once , Read Many times disks can be written only once.
- Very durable and reliable.
- Read Only Disks , such as CD-ROM and DVD come from the factory with the data pre-recorded.

20. What is meant by polling?

The interaction on between the host and a controller can be done using a hand shaking concept. This can be done using the following steps.

1. The host repeatedly reads the bust bit until that bit becomes clear.
2. The host set the write bit in the command register and writes a byte into the data out register.
3. The host sets the command – ready bit.
4. When the controller notices that the command ready but is set, it sets the busy bit.
5. The controller reads the command register and sees the write command. It reads the data –out register to get byte, and does the I/O to the device.
6. The controller clears the command ready bit, clears the error bit in the status register to indicate that the device I/O succeeded, and clears the busy bit to indicate that it is finished.

PART B

16 MARKS QUESTIONS WITH HINTS

UNIT 1

1. Write about the various system calls.

- Process control
- File management
- Device management
- Information maintenance
- Communication

2. Explain about inter process communication.

- Message-passing system
- Naming
- Direct communication
- Indirect communication
- Synchronization
- Buffering

3. Give an overview about threads.

- Thread definition
- Motivation
- Diagram
- Benefits

- User and kernel threads

4. Write about the various CPU scheduling algorithms.

- First-come, first-served scheduling
- Shortest-job-first scheduling
- Priority Scheduling
- Round-robin scheduling
- Multilevel queue scheduling
- Multilevel feedback queue scheduling

UNIT II

1. Explain the classic problems of synchronization.

- The bounded-buffer problem with structure
- The readers-writers problem with structure
- The dining-philosophers problem with structure

2. Write about critical regions and monitors.

- Critical region definition
- Implementation of the conditional-region construct
- Monitor definition
- Syntax of monitor
- Schematic view of monitors
- Monitor with condition variables
- Monitor solution to dining-philosopher problem

3. Give a detailed description about deadlocks and its characterization

- Deadlock definition
- Deadlock conditions
- Mutual exclusion
- Hold and wait
- No pre-emption
- Circular wait
- Resource allocation graph

4. Explain the Banker's algorithm for deadlock avoidance.

- Deadlock avoidance definition
- Data structures used
- Safety algorithm
- Resource request algorithm

UNIT III

1. Write about the techniques for structuring the page table.

- Hierarchical paging-two-level & multi-level with diagram
- Hashed page table with diagram
- Inverted page table with diagram

2. Explain the basic concepts of segmentation.

- User view of program
- Segmentation definition
- Hardware used with diagram-segment table, base, limit & offset
- Protection and sharing with diagram
- Fragmentation

3. What is demand paging and what is its use?

- Demand paging definition
- Virtual memory implementation
- Lazy swapper, page fault, pure demand paging, valid-invalid bit
- Diagrams

4. Explain the various page replacement strategies.

- Page replacement-basic scheme with diagram
- FIFO page replacement
- Optimal page replacement
- LRU page replacement

- LRU approximation page replacement
- Counting-based page replacement
- Page buffering algorithm

UNIT IV

1. What are files and explain the access methods for files?

- File definition
- Attributes, operations and types
- Sequential access with diagram
- Direct access
- Other access methods-index with diagram

2. Explain the schemes for defining the logical structure of a directory.

- Single level directory with diagram
- Two level directory with diagram
- Tree structured directory with diagram
- Acyclic-graph directory with diagram
- General graph directory with diagram

3. Write notes about the protection strategies provided for files.

- Types of access
- Access control list (ACL)
- Three classifications-owner, group & universe
- Other protection approaches-passwords

4. Discuss various file allocation methods.

- Contiguous Allocation
- Linked Allocation
- Indexed Allocation

UNIT V

1. Write about the kernel I/O subsystem.

- I/O scheduling
- Buffering
- Caching
- Spooling & device reservation
- Error handling
- Kernel data structures

2. Explain the various disk scheduling techniques

- FCFS scheduling
- SSTF scheduling
- SCAN scheduling
- C-SCAN scheduling
- LOOK scheduling

3. Write notes about disk management and swap-space management.

- Disk formatting-low level formatting
- Boot block-bootstrap loader, boot block, boot disk & system disk
- Bad blocks-sector sparing, sector slipping
- Swap-space use
- Swap-space location
- Swap-space management

4. Explain the allocation methods for disk space.

- Contiguous allocation advantage, disadvantage & diagram
- Linked allocation advantage, disadvantage & diagram
- Indexed allocation advantage, disadvantage & diagram
- Performance

* * * * * *